Date=7/12/2020
Lecture By= Akash Handa
Subject ⇒ React Introduction

IN PREVIOUS LECTURE **(QUICK RECAP) Date-4/12/2020**	**In Today's Lecture (Overview)**
Redux Authentication Reducer connect() Overview	React Introduction What is React? How does React Work? React.JS History React ES6 What is ES6? Why Should I Learn ES6? Classes The Render Function Example The HTML Code React JSX What is JSX? Coding JSX React Components React Components Create a Class Component Create a Function Component Component Constructor Example

React Introduction

What is React?

React is a JavaScript library created by Facebook.

React is a tool for building UI components.

How does React Work?

React creates a VIRTUAL DOM in memory.

Instead of manipulating the browser's DOM directly, React creates a virtual DOM in memory, where it does all the necessary manipulating, before making the changes in the browser DOM.

React only changes what needs to be changed!

React finds out what changes have been made, and changes only what needs to be changed.

You will learn the various aspects of how React does this in the rest of this tutorial.

React.JS History

Current version of React.JS is V16.8.6 (March 2019).

Initial Release to the Public (V0.3.0) was in July 2013.

React.JS was first used in 2011 for Facebook's Newsfeed feature.

Facebook Software Engineer, Jordan Walke, created it.

The create-react-app version 2.0 package was released in October 2018.

Create-react-app version 2.0 supports Babel 7, webpack 4, and Jest23.

To get an overview of what React is, you can write React code directly in HTML.

But in order to use React in production, you need NPM and Node.js installed.

React ES6

What is ES6?

ES6 stands for ECMAScript 6.

ECMAScript was created to standardize JavaScript, and ES6 is the 6th version of ECMAScript, it was published in 2015, and is also known as ECMAScript 2015.

Why Should I Learn ES6?

React uses ES6, and you should be familiar with some of the new features like:

- Classes
- Arrow Functions
- Variables (let, const, var)

Classes

ES6 introduced classes.

A class is a type of function, but instead of using the keyword `function` to initiate it, we use the keyword `class`, and the properties are assigned inside a `constructor()` method.

Click Below to know more about It
https://www.w3schools.com/react/react_es6.asp

The Render Function

The `ReactDOM.render()` function takes two arguments, HTML code and an HTML element.

The purpose of the function is to display the specified HTML code inside the specified HTML element

Example

Display a paragraph inside the "root" element:

```
ReactDOM.render(<p>Hello</p>, document.getElementById('root'));
```

The result is displayed in the `<div id="root">` element:

```
<body>

  <div id="root"></div>

</body>
```

The HTML Code

The HTML code in this tutorial uses JSX which allows you to write HTML tags inside the JavaScript code:

React JSX

What is JSX?

JSX stands for JavaScript XML.

JSX allows us to write HTML in React.

JSX makes it easier to write and add HTML in React.

Coding JSX

JSX allows us to write HTML elements in JavaScript and place them in the DOM without any `createElement()` and/or `appendChild()` methods.

JSX converts HTML tags into react elements.

Let us demonstrate with two examples, the first uses JSX and the second does not:

JSX:

```
const myelement = <h1>I Love JSX!</h1>;

ReactDOM.render(myelement, document.getElementById('root'));
```

Click below to To know more about it
https://www.w3schools.com/react/react_jsx.asp

React Components

Components are like functions that return HTML elements.

React Components

Components are independent and reusable bits of code. They serve the same purpose as JavaScript functions, but work in isolation and returns HTML via a render function.

Components come in two types, Class components and Function components, in this tutorial we will concentrate on Class components.

Create a Class Component

When creating a React component, the component's name must start with an upper case letter.

The component has to include the `extends React.Component` statement, this statement creates an inheritance to React.Component, and gives your component access to React.Component's functions.

The component also requires a `render()` method, this method returns HTML.

Create a Function Component

Here is the same example as above, but created using a Function component instead.

A Function component also returns HTML, and behaves pretty much the same way as a Class component, but Class components have some additions, and will be preferred in this tutorial.

Example

Create a Function component called `Car`

```
function Car() {
  return <h2>Hi, I am also a Car!</h2>;
}
```

Once again your React application has a Car component.

Refer to the Car component as normal HTML (except in React, components *must* start with an upper case letter):

Component Constructor

If there is a `constructor()` function in your component, this function will be called when the component gets initiated.

The constructor function is where you initiate the component's properties.

In React, component properties should be kept in an object called `state`.

You will learn more about `state` later in this tutorial.

The constructor function is also where you honor the inheritance of the parent component by including the `super()` statement, which executes the parent component's constructor function, and your component has access to all the functions of the parent component (`React.Component`).

Example

Create a constructor function in the Car component, and add a color property:

```
class Car extends React.Component {
  constructor() {
    super();
    this.state = {color: "red"};
  }
  render() {
    return <h2>I am a Car!</h2>;
  }
}
```